

**ОДСЕК ЗА СОФТВЕРСКО ИНЖЕЊЕРСТВО**  
**АЛГОРИТМИ И СТРУКТУРЕ ПОДАТАКА 2**  
**2020-2021**

- други домаћи задатак -

**Опште напомене:**

1. Пре одбране сви студенти раде тест знања који се ради на рачунару коришћењем система Moodle (<http://elearning.rcub.bg.ac.rs/moodle/>). **Сви студенти треба да креирају налог и пријаве се на курс пре почетка лабораторијских вежби.** Пријава на курс ће бити прихваћена и важећа само уколико се студент региструје путем свог налога електронске поште на серверу **mail.student.etf.bg.ac.rs**.
2. Домаћи задатак 2 састоји се од једног програмског проблема. Студенти проблем решавају **самостално**, на програмском језику C++.
3. Реализовани програм треба да комуницира са корисником путем једноставног менија који приказује реализоване операције и омогућава sukcesivnu примену операција у произвољном редоследу.
4. Унос података треба омогућити било путем читања са стандардног улаза, било путем читања из датотеке.
5. Решења треба да буду отпорна на грешке и треба да кориснику пружи јасно обавештење у случају детекције грешке.
6. Приликом оцењивања, биће узето у обзир рационално коришћење ресурса. **Примена рекурзије се неће признати као решење проблема које може освојити максималан број поена.**
7. За све недовољно јасне захтеве у задатку, студенти треба да усвоје разумну претпоставку у вези реализације програма. Приликом одбране, демонстраторе треба обавестити која претпоставка је усвојена (или које претпоставке су усвојене) и која су ограничења програма (на пример, максимална димензија низа и слично). Неоправдано увођење ограничавајуће претпоставке повлачи негативне поене.
8. Одбрана другог домаћег задатка ће се обавити према распореду који ће накнадно бити објављен на сајту предмета. Пријава за одбрану је омогућена преко Moodle система. Детаљније информације су објављене на предметном сајту.
9. Предаја домаћих ће бити омогућена преко Moodle система до **уторка, 01.12.2020. у 23:59**. Детаљније информације су објављене на предметном сајту.
10. За решавање задатака који имају више комбинација користити следеће формуле.  
(**R** – редни број индекса, **G** – последње две цифре године уписа):  
$$i = (R + G) \bmod 2$$
11. Предметни наставници задржавају право да изврше проверу сличности предатих домаћих задатака и коригују освојени број поена након одбране домаћих задатака, као и да пријаве теже случајеве повреде Правилника о дисциплинској одговорности студената Универзитета у Београду Дисциплинској комисији Факултета.

## Задатак 1 – Имплементација приоритетног реда коришћењем 2-3-4 стабала [100 поена]

Приоритетни ред се имплементира коришћењем црвено-црних стабала представљеним помоћу изоморфних **2-3-4 стабала**. Елементи реда су акције представљене стрингом до 256 знакова и њима додељени целобројни приоритети. Уколико се у реду нађу акције са истим приоритетом, оне треба да буду уређене лексикографски. Приоритети акција се временом могу мењати, за сваку акцију посебно. Акције су у стаблу уређене по приоритетима, и у зависности од редног броја проблема  $i$  који се решава:

0. већи број означава виши приоритет
1. мањи број означава виши приоритет

Реализовати следеће операције над 2-3-4 стаблом:

1. **[10 поена]** Стварање празног стабла и уништавање стабла
2. **[20 поена]** Претраживање и уметање
  - Претраживање стабла по приоритету акције
  - Уметање нове акције у стабло
3. **[20 поена]** Брисање акције из стабла
  - Брисање произвољне акције по приоритету
  - Брисање акције највишег приоритета
4. **[10 поена]** Исписивање стабла у излазни ток (оператор  $<<$ )
  - Исписивање 2-3-4 стабла
  - Исписивање у облику бинарног црвено-црног стабла
5. **[15 поена]** Измену приоритета акције, што ефикасније могуће
6. **[15 поена]** Проналажење укупног броја акција са задатим приоритетом
7. **[10 поена]** Главни програм који врши комуникацију са корисником
  - Читање и манипулација појединачним акцијама
  - Читање и формирање стабла на основу података из датотеке

Један ред датотеке садржи име акције (стринг до 256 знакова) и њен приоритет. На основу садржаја датотеке треба омогућити формирање стабла које представља приоритетни ред и даљу манипулацију њиме.

## Напомене

У задатку по потреби реализовати и додатне методе, где је то примерено. Написати главне програме који кориснику омогућавају рад са задатим стаблом, путем једноставног интерактивног менија.

Рад са датотекама у језику C++ захтева увожење заглавља `fstream` (именски простор `std`). За читање података користи се класа `ifstream`. Након отварања датотеке, читање се врши на исти начин као и са стандардног улаза. Кратак преглед најбитнијих метода и пријатељских функција ове класе је дат у наставку.

<pre>void open(     const char *_Filename,     ios_base::openmode _Mode = ios_base::in,     int _Prot = (int)ios_base::_Openprot );</pre>	<p>Отвара датотеку задатог имена за читање.</p> <pre>ifstream dat; dat.open("datoteka.txt");</pre>
<pre>void close();</pre>	<p>Затвара датотеку.</p>
<pre>bool is_open();</pre>	<p>Утврђује да ли је датотека отворена.</p>
<pre>operator&gt;&gt;</pre>	<p>Преклопљен оператор за просте типове података.</p>
<pre>ifstream dat; dat.open("datoteka.dat"); if( ! dat.is_open() )    greska(); char niz[20]; dat &gt;&gt; niz; dat.close();</pre>	<p>Пример отварања датотеке, провере да ли је отварање успешно, читање једног знаковоног низа из датотеке и затварања датотеке.</p>